Modern Applied Science; Vol. 12, No. 4; 2018
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

A Power Saving Hybrid Technique for IoT

Sadeq Al-Hamouz[1]

[1] Computer Science Department, the World Islamic Sciences and Education University, Jordan

Correspondence: Sadeq Al-Hamouz, Computer Science Department, the World Islamic Sciences and Education University, Amman, Jordan. E-mail: Sadeq.Alhamouz@wise.edu.jo

Received: February 27, 2018 Accepted: March 6, 2018 Online Published: March 19, 2018

doi:10.5539/mas.v12n4p30 URL: https://doi.org/10.5539/mas.v12n4p30

Abstract

Most IoT applications usually contain a battery inside, which provides good mobility for the system, but also limits lifetime of the application in general. A lot of work concentrated on power saving algorithms for software, and hardware solutions to provide better results. The proposed solution uses a hybrid technique that enables the IoT system to save power by installing a secondary low current microcontroller to put sensors and other devices in sleep mode, and depending on the IoT device's application, it will be awakened by a needed trigger that is tuned according to the application (such as certain sensors readings, or time based trigger). This will provide high system flexibility.

Keywords: power saving, Internet of Things (IoT), System on Chip (SoC), current consumption, Wi-Fi adapter

1. Introduction

Smart homes and smart devices are taking the attention of consumers everywhere. Surveys show that almost 81% of Americans (Note 1), for example, are interested or already own smart devices for their smart homes. Connecting these devices (or things) through a wireless network to enable them to communicate and be remotely access or controller can be used to define the emerging technology of the Internet of Things (IoT).

The "Things" that are listed for this technology are limitless; they vary from cars to toasters, from the Air Conditioning Unit to the light bulbs (Al-Fuqaha et al., 2015). These things are used in almost every aspect of life, whether it is in homes, small shops or big institutions (Perera et al., 2014).

As with every other new technology, there come some limitations and concerns associated with the wide spread of it. The IoT connected devices are vulnerable to power sources limitations for example, since the chips that are designated for communication with other "things" in an IoT system use power to establish and confirm connections, or to send and receive data.

Most IoT applications usually contain a battery embedded within the device, but this might lead to limiting the lifetime of the application in general. For example, a microcontroller of type ESP8266 with built-in WiFi consumes 170mA while transmitting data, about 15mA in modem sleep mode, and with deep sleep, it consumes 0.01mA only (but software will not be able to wake it up, it needs hardware reset).

The proposed system shows a cutting edge technology for power consumption utilization using a hybrid technique that relies on installing a secondary chip to the original System on Chip (SoC) to enhance power consumption of smart devices in an IoT system.

2. Related Work

There have been many ideas and innovations concerning power saving and management in a IoT system, some focused on hardware integration (System on a Chip, aka SoC) of power sources, while others focused on software that enable devices to sleep and wake up based on certain triggers and conditions.

With around one trillion semiconductors that are used in IoT devices (Karimi and Atkinson, 2013), the need for an efficient power management protocols and devise is highly needed. Some promising technologies were proposed in the form of an Ultra-Low Power (ULP) devices (Myers et al., 2014) and (Rossi et al., 2017) that uses less than 10μW of energy, while being able to use sensors, process collected data, and transmit them (Rossi et al., 2017) , but such systems have limitations since a dedicated software that is efficient and "smart" need to co-exist with these devices, which might result in the opposite intention of adopting this technology; that is increase power consumption.

Some researchers adopted devices that are battery-less, like (Yahya et al., 2017), where a System in Package (SiP) was integrated with a 1Mb/s Radio transmitter and a non-volatile memory, to increase SoC efficiency and reduce cost. The SoCs use a utilized Energy harvesting Platform to supply the SoC and off-chip components in the IoT environment.

Another technique was to adopt self-powered devices, as in the work of (Klinefelter et al., 2015) and (Shirvanimoghaddam et al., 2017) where energy is harvested from the environment and integrated into the SoC of an IoT device to make it functional. The main drawback with such technology is that they are best integrated with large systems, which implies they will not be suitable to certain IoT applications, like in medical monitoring systems (Lian, 2015). Another note relates to the unstable nature of environmental power sources where power might be lost and devices are turned off, some researchers tackled this problem, like (Jayakumar et al., 2017), but data was lost even when power was restored.

Availability of IoT devices should be maintained all the time. Since Wireless Sensor Networks (WSN) are an architecture where IoT devices can be connected and communicate through, the protocols that are used to save power in nodes of such a network are applicable to IoT systems (Kaur and Kaur, 2016), like the commonly known IEEE 802.15.4/Zigbee, Wi-Fi/IEEE 802.11 protocols (Mahmoud and Mohamad2016), or the Bluetooth Low Energy technology (Nieminen et al., 2014) and (Collotta and Pau, 2015).

The use of WSN for IoT system has been extensively studied from energy saving perspective, several techniques were developed to ensure power saving (Kaur and Kaur, 2016), the researchers of (Toklu and Erdem, 2014) recommended the employment of Base Station Controlled MAC (BSC-MAC) that is an emerging adaptive, power saving variation to the original MAC in terms of scheduling sleep turns. Others like (Shahgoshtasbi and Jamshidi, 2014) focused on the Demand response (DR) structure that utilized sum of timing through a Neuro-fuzzy paradigm to save energy and was proven effective given a network with no exceptional cases.

3. System Schematic

Most IoT systems rely on enhancing either the hardware power management efficiency, or the sleep cycle of connected devices scheduling. The system proposed in this research employs a hybrid technique that enhances power management for IoT connected devices.

We suggest in this research embedding a secondary very low current microcontroller (ATTINY84), which consumes <0.5mA on idle mode with NPN transistors (2N2222A) to power off sensors, and main controller when needed (via a specially designed battery saving algorithm), which will reduce full system power consumption from 30 – 80mA (including sensors after applying basic power saving algorithms) to 0.2mA after applying the proposed hybrid hardware/software technique.

The system's main idea concentrates on adding a secondary, very low power, microcontroller to the original NPN transistor that acts as a switch to power-off sensors when in idle state. This technique helps in saving around 99.3% of overall power consumption.

3.1 The Hardware

The secondary microcontroller that is used as a backup for the original SoC is of type Arduino ATTINY84, and has very low power consumption at various frequencies. The current consumed with mostly used frequencies are listed in table 1.

Table 1. ATTINY 84 current consumption (at 3.3V power supply)

Frequency	Current consumption
12 MHz	4 mA
10 MHz	3.8 mA
8 MHz	3 mA
4 MHz	2 mA

The system was developed using a robot that contains ESP8266 microcontroller, motor actuators, motion sensor, LDR sensor, microwave sensor (type MO9 402), GPS unit (type Ublox NEO-6M) and a power source (Li-ion battery 3.7V 3000mA). Table 2 shows information about sensors/actuators used by the developed robot.

Table 2. Robot has used sensors/actuators

Sensor	Current consumption	Number of samples needed	Time needed to measure sample
Motion	15 mA	10 samples/second	2 mS
LDR	1 mA	1 sample/second	10 mS
Microwave	300 mA	1 sample/second	1 mS
DC motors	2 A	--	--
GPS	50 mA	--	10 S

The microcontroller unit used in this research (ESP8266) is popular in power saving systems (like the system developed in (Ram and Gupta, (2016)) because of it's easy to configure sleep modes. These modes are (EPS user guide, online resource):

1. Modem sleep: where system CPU and other internal hardware keeps running, but Wi-Fi modem circuit is powered off while keeping connection up with router but without any data transmission. The processor wakes up Wi-Fi circuit every pre-determined time period (say every 300 mS) to check and receive any packets that were waiting while in modem sleep, then goes back to sleeps. The Processor checks sensors readings while modem is sleeping.

2. Light sleep: which is similar to Modem sleep but includes powering off CPU and modem together. This means less energy consumption by the system, but, on the other hand, the CPU is unable to monitor sensors. This method may be used in insensitive applications and saves around 13mA more than modem sleep method

3. Deep sleep: where the Wi-Fi connection is not maintained, and the unit powers off both modem, and CPU. System wakes up every time period (much longer than modem sleep periods, like every five minutes) to check sensors readings and send it to Wi-Fi then sleeps again. This concept is not applicable in smart cities (that employ IoT) but it uses <1 mA on average.

3.2 The Algorithm

Depending on application, background algorithm in the ATTINY84 chip will initialize a timer interrupt depending on the highest sensor frequency. For example, motion sensor is being read 10 times a second while LDR, and microwave are being read 1 time a second. Thus, timers interrupt need to be set every 100mS (depending on highest frequency of 10Hz).

Inside the interrupt, if it is time to get a sensor's reading, it has switched on via NPN transistor, then read (reading time is same as measurement time plus time needed to stabilize the sensor). After all readings are taken, the serial port with ESP8266 is checked for any incoming data like instant need of sensors readings (as user command) or motor movement to specific position.

To achieve interrupt function, every time an interrupt occurs, motion sensor is being read, and a counter is increased by 1. Once the counter reaches 10, LDR, and microwave sensors are read in the interrupt, and the counter is cleared to 0 again, and so on.

In the main subroutine, a comparison between sensors readings takes place, and depending on user value set, microcontroller may decide to send these value readings (depending on timer, like send data readings every X seconds, or depending on action such as sensor abnormal activity).

Every time ESP8266 wakes up to check for new data receive, it will also check incoming serial connection for new sensors readings to send to cloud storage. Only one node will be listening to incoming packets in the system (chosen depending on power and RSSI signal connection to server) while other nodes Wi-Fi is off. If there is an incoming command to send sensors readings instantly, ESP8266 sends that command to ATTINY84 via serial and waits to receive sensors readings to send them back to user before sleeping again. The chart in figure 1 illustrates the interrupt event handling against time for the example mentioned above.

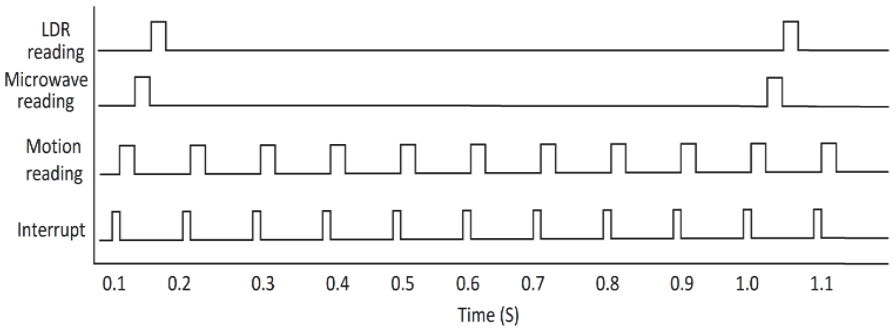

Figure 1. ATTINY84 microcontroller interrupt event handling vs time

Once node in charge of listening receives information, it will either forward them to the destination node directly (when the packet's size is small), or indirectly (when the packet's data is big or the data needs special permission to get access), or may decide to wake up all nearby nodes (if it is a firmware update packet or urgent broadcast packet).

4. The Case Study

A military robotic node in an IoT system is used to receive commands to actuate motors, and move to a specific GPS location to get data from, and, when Robot reaches the spot, it will start monitoring motion, LDR, sound and microwave sensors, and send information to base station every 5 minutes on normal state, or when a significant reading is received from any sensor. The Robot should also maintain a Wi-Fi connection with a router so that it moves immediately to new locations when needed.

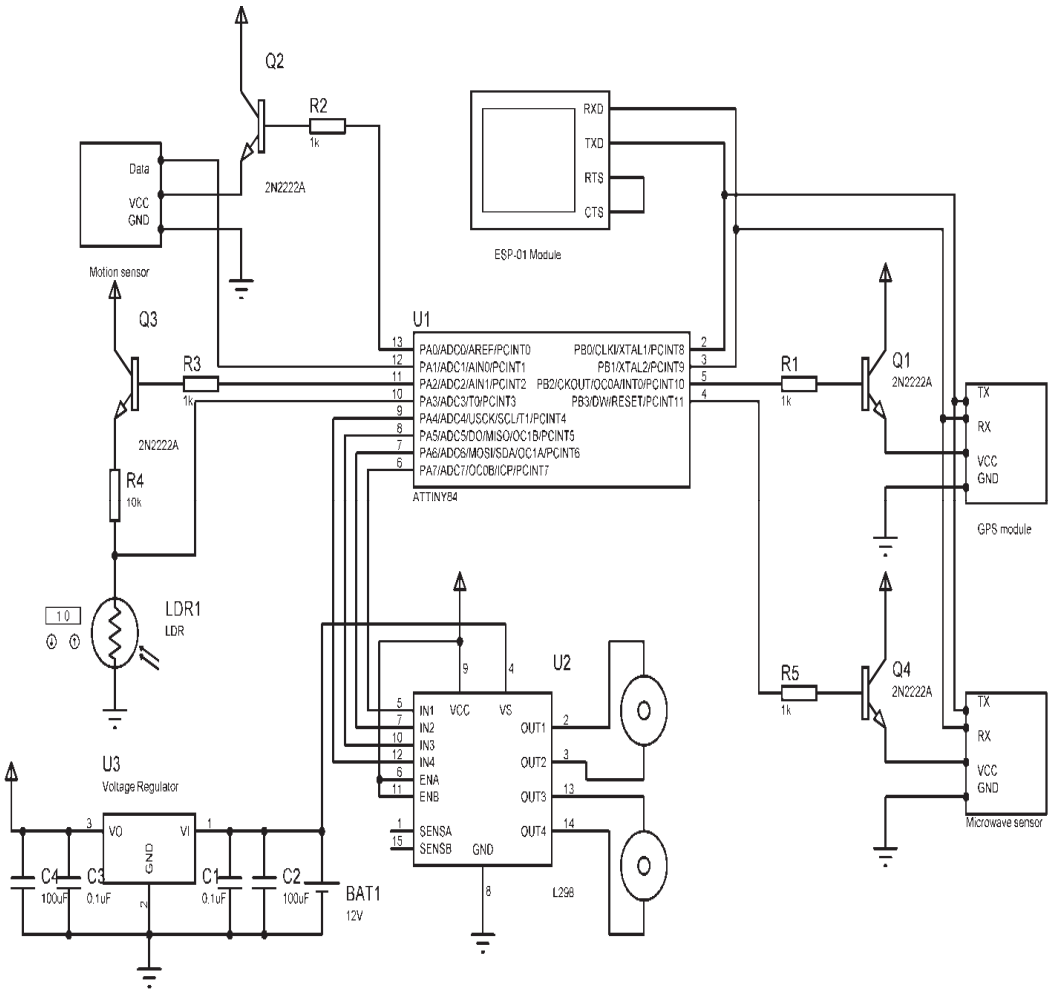

Figure 2. Proposed system connection in the case study

Usually; ESP microcontroller unit is installed and kept in modem sleep mode consuming 14mA and all sensors are

active for reading except the GPS unit (stays in sleep mode, consuming 10mA), where the total sensor power consumption is 326 mA (motors are not active). Thus, total ESP and sensors consume 340mA. Using a 3000mA battery, the system will run for about 8 hours 48 minutes.

In the proposed system, the chips basically depend on having two microcontroller units, a main unit which is ESP8266 and a secondary controller which is ATTINY84. The ESP unit will be in light sleep mode (but could be altered depending on application requirement) consuming 1mA, and ATTINY will be powered on using internal 4 MHz crystal and consuming 2mA. Sensors will be connected to ATTINY, which uses low current, and will be fully monitored by it. When needed (like receiving a reading from one of the sensors), ATTINY will send an interrupt signal to ESP unit causing it to wake up and send information via Wi-Fi. The systems hardware architecture and hardware connections is illustrated in figure 2.

In the system's connection, multiple NPN transistors were used to power off sensors when not needed, besides, same Serial (Tx, Rx) ports are used to communicate with ESP8266, GPS and Microwave sensors. It is known that only one device can communicate through serial at a given point of time, but since GPS and Microwave sensors can be turned off via NPN transistors, ATTINY84 microcontroller can cooperate via code on what device are open for communication and turn off other devices to prevent interference. By the time ATTINY84 microcontroller is trying to communicate with GPS or microwave sensors, ESP8266 will disable serial port via software code to prevent interference (because ESP8266 is an essential unit and can't be turned off via transistor, thus, its ports need to be disabled via code).

At this time, ESP will only monitor Wi-Fi to see if there is any incoming packet with a command to move to another location, in this case, ESP will send new location to ATTINY, which will not keep sensors on all the time, but will only power them on for a certain period of time to read data then power them off again using NPN transistors. This will make power consumption even less as shown in table 3.

Table 3. Different sensors power consumption rates (EPS user guide, online resource)

Sensor	Working time per second	Sensor working rate	Current consumption
Motion	20 mS	2%	300 µA
LDR	10 mS	1%	10 µA
Microwave	1 mS	0.1%	300 µA
GPS	--	--	50 µA (estimated)

To show the strength points of the proposed system, we compared the values registered for power consumption with three different applications that used a somewhat similar technique to the one used in this research.

The authors of (Gia et al., 2018) developed a wearable device in an IoT system, where small chip was embedded inside the smart device to monitor falls for elderly people. While the authors in (Lombardo et al., 2012) focused the attention in power saving on the WiFi adapter in a Wireless Sensor Network, where an ATTINY 2313V chip was installed within an FPGA system to monitor and control energy consumption. In the research (Ignjatovic et al., 2012) the ATTINY 2313 microchip was used also for power management in an IoT system that monitors air filters performance and power consumption.

Table 4. Current consumption of ESP8266 microconteoller depending on practical measurement (3.3V power supply)

State	Current consumption
Data transmission over WiFi	162 mA
Data receiving over WiFi	61 mA
Modem sleep	14 mA
Light sleep	1 mA
Deep sleep	12 uA
Power off	0.2 uA

5. Testing Results

After running the developed robot that is equipped with the aforementioned Chips, the total sensor power consumption was registered to be around 660µA, and both controllers consumed 3 mA, which summed up to a total

of 3.66 mA, compared to the originally needed 340 mAs, the 13.1mA recorded in the work of (Gia et al., 2018), and the 31.65mA consumed in the research in (Lombardo et al., 2012).

The systems that are used for comparison show data for the WiFi adapter only (with the ESP chip installed). The proposed system of this research has the following data listed in table 4 for the WiFi adapter at different states to facilitate relating numbers.

This is a strong indication that the system's working time per charge went up from 8 hours and 48 minutes to 34 days (816 hours), while the similar technique that was developed in (Gia et al., 2018) made the system operable for 228 hours (9 days and 12 hours), and 106 hours (4 days and 5 hours) for the controlling system in (Ignjatovic et al., 2012). Figure 3 shows the trend of power consumption of the proposed system, while the trend of the power saving technique mentioned in (Gia et al., 2018) is plotted in figure 4, and the trend for the Wireless Filters Monitoring System (WFMS) of power consumption is shown in figure 5.

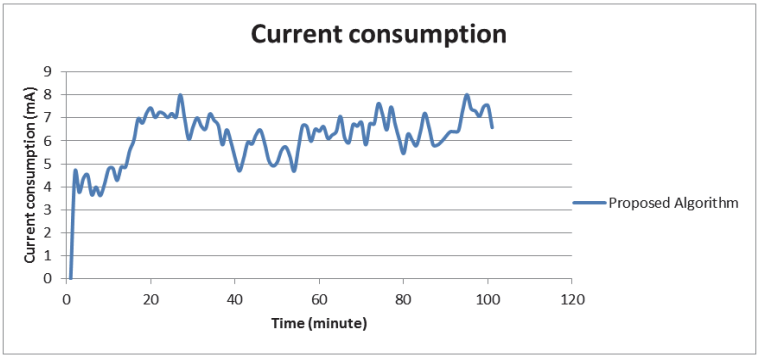

Figure 3. Power consumption trend of the proposed hybrid technique

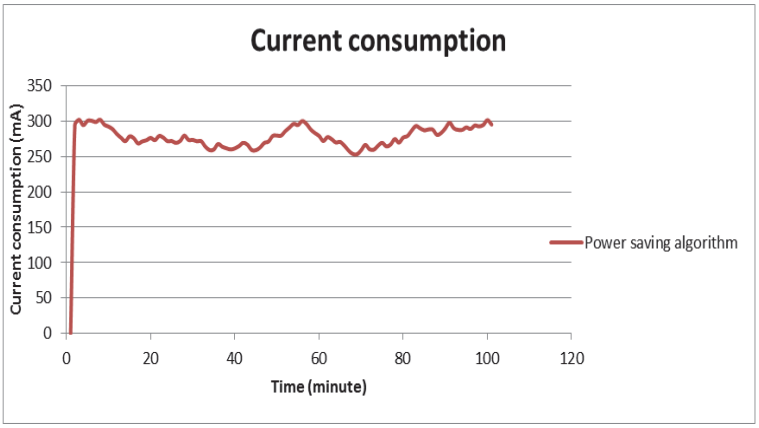

Figure 4. Power consumption trend of the power saving technique in (Gia et al., 2018)

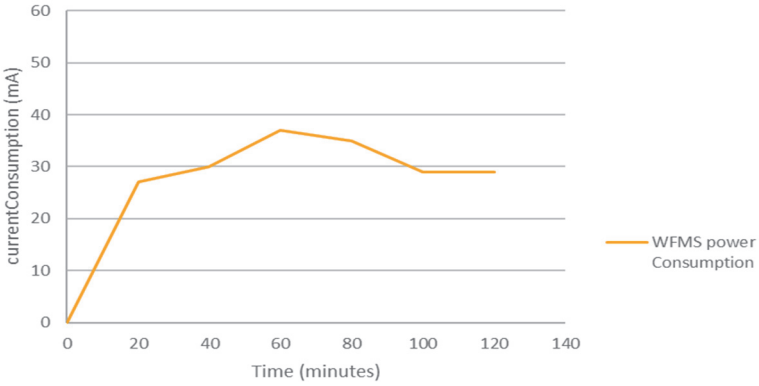

Figure 5. Power consumption trend of the power saving technique in (Ignjatovic et al., 2012)

A brief comparison of the proposed work in this research with the related work mentioned in (Gia et al., 2018),

(Lombardo et al., 2012), and (Ignjatovic et al., 2012) is summarized in table 5

Table 5. Comparison between power consumption between proposed system and other systems from literature

Mode	Elderly Fall detection (Gia et al., 2018)	FPGA for WSN (Lombardo et al., 2012)	WFMS (Ignjatovic et al., 2012)	Proposed System
Sleep mode	8 µA	0.7 mA	27 mA (listening mode-has no sleep mode)	12 µA
Transmitting Mode	32.61 mA	165.58 mA	35 mA	162 mA

The Proposed algorithm proved to be very efficient and can increase battery life (while keeping same system performance) to up to 98%. Though we believe the comparison might not be fair, given that we are comparing the overall performance of the system (where the WiFi adapter is only one of several parts of the system), yet the comparison numbers were in favor of the developed system of this research.

6. Conclusion

There is an undeniable need for an efficient power management scheme to accompany the widely spreading use of IoT in smart homes and institutions. Most devices need to stay up almost at all-time to ensure the benefits of IoT systems are met.

A hybrid technique was proposed in this research that makes power consumption as minimum as possible, while making the smart devices always available. The technique depends on installing and additional chip with very small power consumption to the already operating SoC installed in most smart devices in an IoT system to regulate the sleep modes cycle of connected devices.

The performance and testing of the innovation shows huge saving in power, which lead to prolonging the battery life and reduced the number of recharging times needed, thus enhancing the overall efficiency of the entire IoT system.

References

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015) Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials, 17*(4), 2347-76. https://doi.org/10.1109/COMST.2015.2444095

Collotta, M., & Pau, G. (2015). Bluetooth for Internet of Things: A fuzzy approach to improve power management in smart homes, *Computers & Electrical Engineering, 44*, 137-152. https://doi.org/10.1016/j.compeleceng.2015.01.005

EPS user guide. (2018). Retrieved January 10, 2018, from http://www.espressif.com/sites/default/files/9b-esp8266-low_power_solutions_en_0.pdf

Gia, T. N., Sarker, V. K., Tcarenko, I., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2018), Energy efficient wearable sensor node for IoT-based fall detection systems. *Microprocessors and Microsystems, 56*, 34-46. https://doi.org/10.1016/j.micpro.2017.10.014

Ignjatovic, I., Šešlija, D., Tarjan, L., & Dudic, S. (2012). Wireless sensor system for monitoring of compressed air filters. *Journal of Scientific and Industrial Research*, 0975-1084.

Jayakumar, H., Raha, A., Stevens, J. R., & Raghunathan, V. (2017). Energy-Aware Memory Mapping for Hybrid FRAM-SRAM MCUs in Intermittently-Powered IoT Devices. *ACM Transactions on Embedded Computing Systems (TECS), 16*(3), 65. https://doi.org/10.1145/2983628

Karimi, K., & Atkinson, G. (2013). What the Internet of Things (IoT) needs to become a reality. White Paper, *FreeScale and ARM*. pp:1-6.

Kaur, K., & Kaur, K. (2016). A study of power management techniques for Internet of Things (IoT). In Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on 2016 Mar 3 (pp. 1781-1785). IEEE. https://doi.org/10.1109/ICEEOT.2016.7754992

Klinefelter, A., Roberts, N. E., Shakhsheer, Y., Gonzalez, P., Shrivastava, A., Roy, A., Craig, K., Faisal, M., Boley, J., Oh, S., & Zhang, Y. (2015). 21.3 A 6.45 µW self-powered IoT SoC with integrated energy-harvesting

power management and ULP asymmetric radios. *In Solid-State Circuits Conference-(ISSCC),* 2015 IEEE International 2015 Feb 22 (pp. 1-3). IEEE. https://doi.org/10.1109/ISSCC.2015.7063087

Lian, Y. (2015). Challenges in the design of self-powered wearable wireless sensors for healthcare Internet-of-Things. *In ASIC (ASICON),* 2015 IEEE 11th International Conference on 2015 Nov 3 (pp. 1-4). IEEE. https://doi.org/10.1109/ASICON.2015.7517022

Lombardo, M., Camarero, J., Valverde, J., Portilla, J., de la Torre, E., & Riesgo, T. (2012). Power management techniques in an FPGA-based WSN node for high performance applications. *In Reconfigurable Communication-centric Systems-On-Chip (ReCoSoC),*7th International Workshop on 2012 Jul 9 (pp. 1-8). IEEE. https://doi.org/10.1109/ReCoSoC.2012.6322888

Mahmoud, M. S., & Mohamad, A. A. (2016), A study of efficient power consumption wireless communication techniques/modules for internet of things (IoT) applications. *Advances in Internet of Things, 22, 6*(02), 19. https://doi.org/10.4236/ait.2016.62002

Myers, J., Savanth, A., Howard, D., Gaddh, R., Prabhat, P., & Flynn, D. (2015). 8.1 An 80nW retention 11.7 pJ/cycle active subthreshold ARM Cortex-M0+ subsystem in 65nm CMOS for WSN applications. *In Solid-State Circuits Conference-(ISSCC),* 2015 IEEE International, (pp. 1-3). IEEE. https://doi.org/10.1109/ISSCC.2015.7062967

Nieminen, J., Gomez, C., Isomaki, M., Savolainen, T., Patil, B., Shelby, Z., Xi, M., & Oller, J. (2014), Networking solutions for connecting bluetooth low energy enabled machines to the internet of things. *IEEE network. 2014 Nov., 28*(6), 83-90. https://doi.org/10.1109/MNET.2014.6963809.

Perera, C., Liu, H., Jayawardena, S., & Chen, M. (2014). A survey on Internet of Things from industrial market perspective, *IEEE Access*, 2, 1660-1679. https://doi.org/10.1109/ACCESS.2015.2389854

Ram, K. S., & Gupta, A. N. (2016). IoT based Data Logger System for weather monitoring using Wireless sensor networks. *International Journal of Engineering Trends and Technology (IJETT), 2016*, 32. https://doi.org/10.1109/JSYST.2013.2291943

Rossi, D., Loi, I., Pullini, A., & Benini, L. (2017), Ultra-Low-Power Digital Architectures for the Internet of Things. *In Enabling the Internet of Things 2017* (pp. 69-93). Springer International Publishing. https://doi.org/10.1007/978-3-319-51482-6_3

Shahgoshtasbi, D., & Jamshidi, M. M. (2014) A new intelligent Neuro–Fuzzy paradigm for energy-efficient homes. *IEEE Systems Journal, 8*(2), 664-73.

Shirvanimoghaddam, M., Shirvanimoghaddam, K., Abolhasani, M. M., Farhangi, M., Barsari, V. Z., Liu, H., Dohler, M., & Naebe, M. (2017). Paving the Path to a Green and Self-Powered Internet of Things. *arXiv preprint arXiv*:1712.02277.

Toklu, S., & Erdem, O. A. (2014). BSC-MAC: energy efficiency in wireless sensor networks with base station control. *Computer Networks, 11*(59), 91-100. https://doi.org/10.1016/j.bjp.2013.12.012

Yahya, F., Lukas, C. J., Breiholz, J., Roy, A., Patel, H. N., Liu, N., Chen, X., Kosari, A., Li, S., Akella, D., & Ayorinde, O. (2017). A battery-less 507nW SoC with integrated platform power manager and SiP interfaces. *In VLSI Circuits, 2017 Symposium*, (pp. C338-C339). IEEE. https://doi.org/10.23919/VLSIC.2017.8008532

Note

Note 1. Source: http://www.telecompetitor.com/survey-81-own-or-have-interest-in-smart-home-devices/ (Accessed on January 10, 2018).